核心技术

**1. Java编程技术**

Java编程技术是大数据学习的基础，Java是一种强类型语言，拥有极高的跨平台能力，可以编写桌面应用程序、Web应用程序、分布式系统和嵌入式系统应用程序等，是大数据工程师最喜欢的编程工具，因此，想学好大数据，掌握Java基础是必不可少的!

**2.Linux命令**

对于大数据开发通常是在Linux环境下进行的，相比Linux操作系统，Windows操作系统是封闭的操作系统，开源的大数据软件很受限制，因此，想从事大数据开发相关工作，还需掌握Linux基础操作命令。

**3、Hadoop**

Hadoop是大数据开发的重要框架，其核心是HDFS和MapReduce，HDFS为海量的数据提供了存储，MapReduce为海量的数据提供了计算，因此，需要重点掌握，除此之外，还需要掌握Hadoop集群、Hadoop集群管理、YARN以及Hadoop高级管理等相关技术与操作!

**4.Hive**

Hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务进行运行，十分适合数据仓库的统计分析。对于Hive需掌握其安装、应用及高级操作等。

**Avro与Protobuf**

Avro与Protobuf均是数据序列化系统，可以提供丰富的数据结构类型，十分适合做数据存储，还可进行不同语言之间相互通信的数据交换格式，学习大数据，需掌握其具体用法。

**HBase**

HBase是一个分布式的、面向列的开源数据库，它不同于一般的关系数据库，更适合于非结构化数据存储的数据库，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统，大数据开发需掌握HBase基础知识、应用、架构以及高级用法等。

**Flume**

Flume是一款高可用、高可靠、分布式的海量日志采集、聚合和传输的系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据;同时，Flume提供对数据进行简单处理，并写到各种数据接受方(可定制)的能力。大数据开发需掌握其安装、配置以及相关使用方法。

**SSM**

SSM框架是由Spring、SpringMVC、MyBatis三个开源框架整合而成，常作为数据源较简单的web项目的框架。大数据开发需分别掌握Spring、SpringMVC、MyBatis三种框架的同时，再使用SSM进行整合操作。

**9.Kafka**

Kafka是一种高吞吐量的分布式发布订阅消息系统，其在大数据开发应用上的目的是通过Hadoop的并行加载机制来统一线上和离线的消息处理，也是为了通过集群来提供实时的消息。大数据开发需掌握Kafka架构原理及各组件的作用和使用方法及相关功能的实现!

**10.Scala**

Scala是一门多范式的编程语言，大数据开发重要框架Spark是采用Scala语言设计的，想要学好Spark框架，拥有Scala基础是必不可少的，因此，大数据开发需掌握Scala编程基础知识!

**11.Spark**

Spark是专为大规模数据处理而设计的快速通用的计算引擎，其提供了一个全面、统一的框架用于管理各种不同性质的数据集和数据源的大数据处理的需求，大数据开发需掌握Spark基础、SparkJob、Spark RDD、spark job部署与资源分配、Spark shuffle、Spark内存管理、Spark广播变量、Spark SQL、Spark Streaming以及Spark ML等相关知识。我这有Spark的学习资料，需要的加好友：2915541363，免费领取。

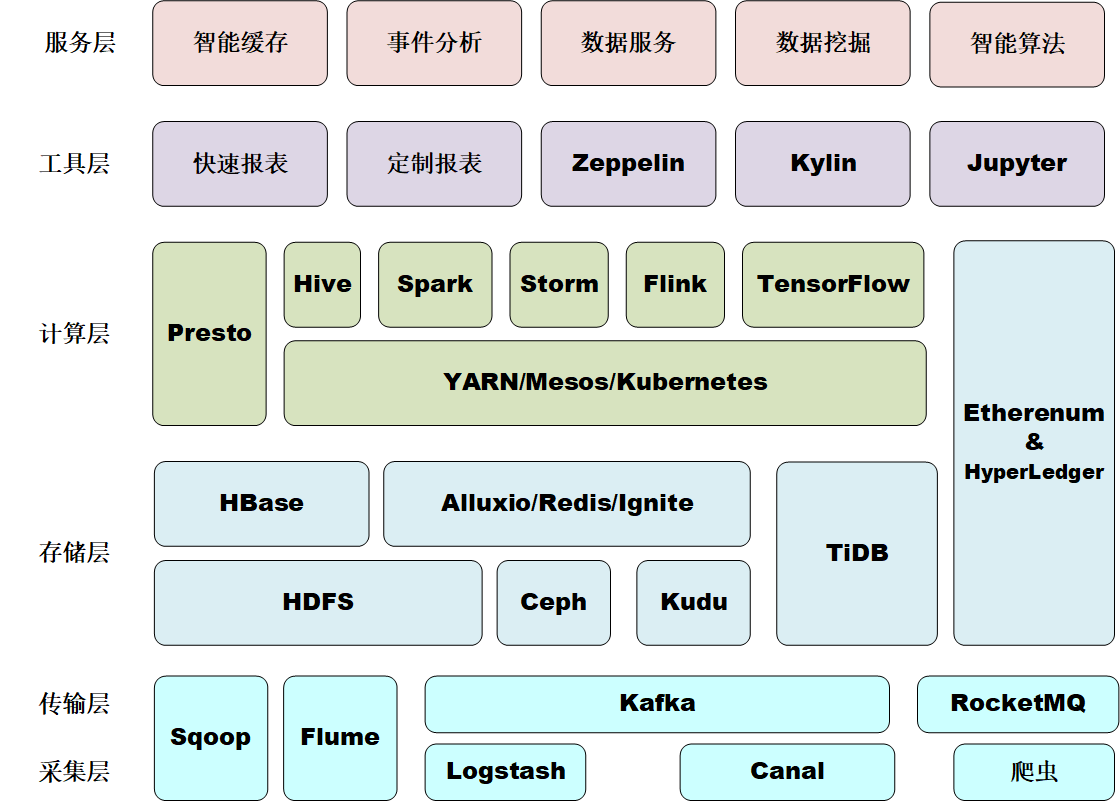
**12.Azkaban**

Azkaban是一个批量工作流任务调度器，可用于在一个工作流内以一个特定的顺序运行一组工作和流程，可以利用Azkaban来完成大数据的任务调度，大数据开发需掌握Azkaban的相关配置及语法规则。

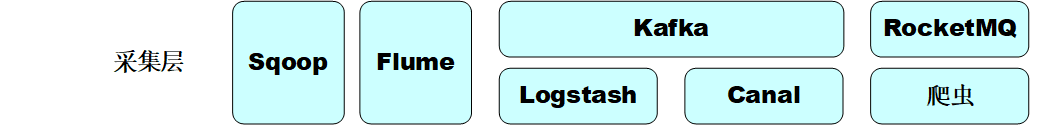
**13.Python与数据分析**

Python是面向对象的编程语言，拥有丰富的库，使用简单，应用广泛，在大数据领域也有所应用，主要可用于数据采集、数据分析以及数据可视化等，因此，大数据开发需学习一定的Python知识。

大数据技术栈全貌



采集层和传输层

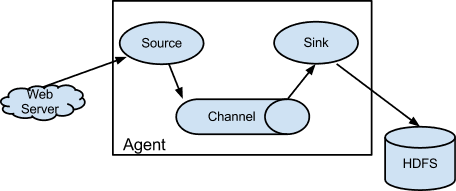


[Sqoop](http://sqoop.apache.org/)

在hadoop和关系型数据库之间转换数据。

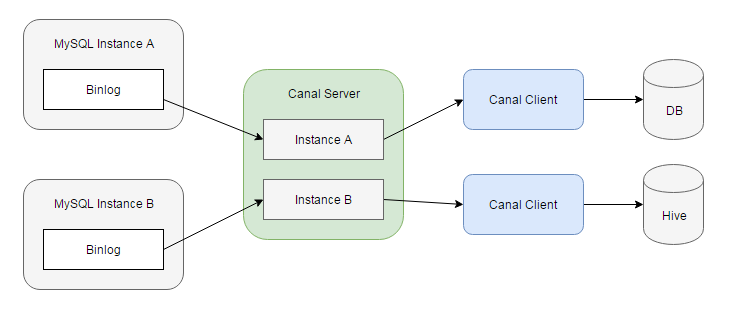
[Flume](http://flume.apache.org/)

Flume是一个分布式的高可用的数据收集、聚集和移动的工具。通常用于从其他系统搜集数据，如web服务器产生的日志，通过Flume将日志写入到Hadoop的HDFS中。



[Canal](https://github.com/alibaba/canal)

数据抽取是 ETL 流程的第一步。我们会将数据从 RDBMS 或日志服务器等外部系统抽取至数据仓库，进行清洗、转换、聚合等操作。



[Logstash](https://www.elastic.co/cn/products/logstash)

Logstash 是开源的服务器端数据处理管道，能够同时从多个来源采集数据，转换数据，然后将数据发送到您最喜欢的 “存储库” 中。

[Kafka](http://kafka.apache.org/)

消息队列，一个分布式流平台。

[RocketMQ](http://rocketmq.apache.org/)

阿里巴巴开源的消息队列。

存储层

[**HBase**](http://hbase.apache.org/)

HBase is the Hadoop database, a distributed, scalable, big data store.

**Alluxio/Redis/Ignite**

[Alluxio](http://www.alluxio.org/)以内存为中心分布式存储系统，从下图可以看出， Alluxio主要有两大功能，第一提供一个文件系统层的抽象，统一文件系统接口，桥接储存系统和计算框架；第二通过内存实现对远程数据的加速访问。

[Redis](https://redis.io/)是一个开源的**内存**键值数据库，相比于Memcache，支持丰富的数据结构。   
[Ignit](https://ignite.apache.org/)是一个**以内存为中心**的分布式数据库，缓存和处理平台，用于事务，分析和流式工作负载，在PB级别的数据上提供接近内存速度访问数据。

**TiDB**

TiDB私有PingCap开源的分布式NewSQL关系型数据库。NewSQL数据库有两个流派，分别是以Google为代表的Spanner/F1和以Amazon 为代表的Aurora(极光)，目前国内做NewSQL数据库主要是参考Google的Spanner架构，Google Spanner也是未来NewSQL的发展趋势。

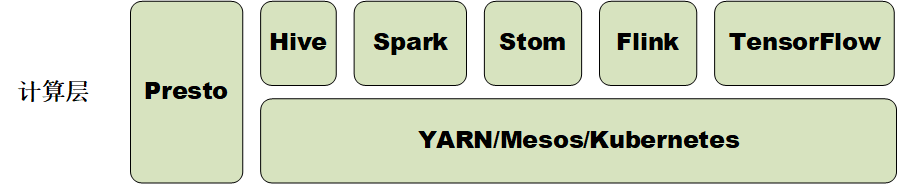
[**HDFS**](http://hadoop.apache.org/)

Hadoop的分布式文件系统。

[**Kudu**](https://kudu.apache.org/overview.html)

Kudu是cloudera开源的运行在hadoop平台上的列式存储系统,拥有Hadoop生态系统应用的常见技术特性，运行在一般的商用硬件上，支持水平扩展,高可用，目前是Apache Hadoop生态圈的新成员之一（incubating）。

计算层



**Hive**

Facebook 开源。Hive是一个构建在Hadoop上的数据仓库框架。Hive的设计目标是让精通SQL技能但Java编程技能相对较弱的分析师能对存放在Hadoop上的大规数据执行查询。

Hive的查询语言HiveQL是基于SQL的。任何熟悉SQL的人都可以轻松使用HiveSQL写查询。和RDBMS相同，Hive要求所有数据必须存储在表中，而表必须有模式（Schema），且模式由Hive进行管理。

**Spark**

Spark是一个分布式计算框架。

**Storm**

Storm是一个分布式的、高容错的实时计算系统。Storm对于实时计算的的意义相当于Hadoop对于批处理的意义。Hadoop为我们提供了Map和Reduce原语，使我们对数据进行批处理变的非常的简单和优美。同样，Storm也对数据的实时计算提供了简单Spout和Bolt原语。

Storm适用的场景：①、流数据处理：Storm可以用来用来处理源源不断的消息，并将处理之后的结果保存到持久化介质中。②、分布式RPC：由于Storm的处理组件都是分布式的，而且处理延迟都极低，所以可以Storm可以做为一个通用的分布式RPC框架来使用。

**Flink**

Apache Flink is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Flink has been designed to run in all common cluster environments, perform computations at in-memory speed and at any scale.

**TensorFlow**

TensorFlow™ is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google’s AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.